

REMARKS

Prior to this amendment, Claims 1-45 were pending in the application. By this amendment, Claims 1, 9-11, 13, 14, 16-20, 23, 24, and 26 are amended. No claims are added or cancelled. Hence, Claims 1-45 are currently pending in the application.

SUMMARY OF THE REJECTIONS/OBJECTIONS

Claim 22 was rejected under 35 U.S.C. §112, second paragraph, as allegedly indefinite.

Claims 1-14, 16-20, 22, 24-37, and 39-43 were rejected under 35 U.S.C. §102(e) as allegedly anticipated by Ng et al. (“Ng”; U.S. Patent No. 6,385,618); Claims 15, 21, 38, and 44 were rejected under 35 U.S.C. §103(a) as allegedly unpatentable over Ng in view of Bennett (“Bennett”; U.S. Patent No. 6,014,733); and Claims 23 and 45 were rejected under 35 U.S.C. §103(a) as allegedly unpatentable over Ng in view of Curtis et al. (“Curtis”; U.S. Patent No. 6,336,216).

THE REJECTIONS NOT BASED ON THE PRIOR ART

Claim 22 was rejected under 35 U.S.C. §112, second paragraph, as allegedly indefinite. Specifically, Claim 22 is rejected due to insufficient antecedent basis for the term “said first class.”

Claim 1 is amended to recite “said data item is an instance of a first class.” Claim 22 depends from Claim 1. Therefore, the term “said first class” in Claim 22 now has sufficient antecedent basis, and this rejection is overcome.

THE REJECTIONS BASED ON THE PRIOR ART

Rejection under 35 U.S.C. §102(e)

Claims 1-14, 16-20, 22, 24-37, and 39-43 were rejected under 35 U.S.C. §102(e) as allegedly anticipated by Ng. These rejections are traversed.

(A) Overview of General State of the Art and Various Embodiments of the Invention

It is thought that an overview of classic object-oriented programming and various embodiments of the invention, and combinations thereof, will assist in understanding how the disclosure of *Ng* differs from embodiments claimed in the application. This overview does not further limit any given claim beyond the language recited in the claim and is, accordingly, to be regarded in an illustrative rather than a restrictive sense.

As stated in the Background section, with classic object oriented programming, inheritance cannot be used (a) to provide different implementations for a particular method for instances from the same class or (b) to provide different attributes for instances from the same class. Thus, because classic inheritance does not allow for instances of the same class to behave differently or have different attributes, there is a need for per-instance methods and attributes.

The application describes mechanisms for enabling per-instance methods and attributes. A mechanism for implementing per-instance attributes involves using “categories,” where an object inherits properties associated with a category class when the object is associated with a corresponding category object (page 8, lines 20-22). In other words, a category class provides a mechanism for categorizing an object (page 10, line 9). A purpose of the category mechanism is to offer another level of flexibility over and above what multiple inheritance offers. As such, the category mechanism allows for associating properties (i.e., methods and attributes) with one instance of a “main” class without having to associate the same properties with any other instances of the same “main” class.

This category mechanism is different than conventional class inheritance, wherein all objects of the same class inherit the same properties from the class and/or subclass from which the objects are instantiated. Thus, with conventional class inheritance, all objects of the same

class are identical in their properties, with the exception of particular values for attributes and/or arguments for methods that may vary from object to object.

(B) How *Ng* Differs From the Claims

Claim 1 recites, *inter alia*, associating a data item, which is an instance of a first class and thus inherits the properties of the first class, with a category object that is an instance of a category class. The category class is a different class from the first class. Significantly, the data item is associated with the category object without associating the category object with all other instances of the first class. Consequently, the data item is now associated with a data structure that includes storage for values for one or more attributes of the category class. A practical result of this is that one object of the first class can be associated with the category object and, therefore, inherit properties of the category class, whereas another object of the same first class is not associated with the category object and does not inherit the properties of the category class.

Reference to Fig. 6 shows that the eight “user defined subclasses” (212, 213, 214, 215, 216, 217, 218 and 219) introduce attributes as depicted in the individual boxes of the eight “user defined subclasses” of the category class 211. Therefore, objects or instantiations of the public_object class 210 (i.e., a “superclass”) may be placed in any number of categories. Figure 7a is an example of a document object, which is on a basic CD format, and where an internal review will be performed on the document. Therefore, the document object 700 (an instance of document class 220) in question is placed in the basic CD category, using basic CD object 701, and the internal review category, using the internal review object 702. Placing document object 700 into the basic CD and internal review categories is accomplished by associating category objects 701 and 702, which are instances of classes 218 and 216 respectively, with document object 700.

In figure 7b, the two category objects (1) basic CD object 701 and (2) internal review object 702 inherit all the properties of the respective superclasses of objects 700, 701, and 702. In addition, objects 701 and 702 inherit all the properties of the respective user-defined category classes, Basic CD 218 and Internal Review 216.

In the rejection of Claim 1, the Office Action relies on the following:

- (1) an “order” object for a teaching of the “data item” of Claim 1;
- (2) a “customer” object and class for a teaching of the category object and class of Claim 1; and
- (3) a foreign key “Customer_for_Order” in Class Order, which references particular customer objects for particular order objects, for a teaching of “associating said data item (e.g., the “order” object) with said category object (e.g., the “customer” object) without associating said category object with all other instances of said first class (e.g., the “order” class) thereby causing said data item to be associated with a structure that includes storage for values for said one or more attributes (e.g., attributes of the category class: “customer” class).”

One feature of Claim 1 that is missing from Ng is the data item (e.g., the “order” object) being associated with a structure that includes storage for values for said one or more attributes of the category class (i.e., inherits the attributes from the category class). For such a teaching to support a rejection of Claim 1, Ng would have to teach that the “order” object inherits the attributes of the “customer” class, which are Cust_id and SSN. Ng does not teach or suggest this. Note that the Cust_id and SSN attributes of the Class Customer are not included or associated with the Class Order.

As mentioned, the “Customer_for_Order” in Class Order is a foreign key that references (e.g., points to) particular customer objects (foreign) that are associated with particular order objects (primary). In other words, for a given order in an “order” table, the foreign key points to the row identifying the customer, in the “customer” table, that made that order. A foreign key relationship between class attributes is simply not the same as inheriting the attributes in an object oriented programming sense. Again, Ng would have to teach that the structure of an “order” object includes storage for values for Cust_id and SSN, which it does not.

Furthermore, Ng does not teach or suggest that the data item (e.g., the “order” object) is associated with the category object (e.g., the “customer” object) without associating said category object with all other instances of the first class (e.g., the “order” class). By contrast, based on Fig. 4A and associated text of Ng, all “order” objects from Class Order must include the field, Customer_for_Order, to implement the foreign key reference to the particular “customer” object that represents the customer that placed the order that a given “order” object represents.

Based on the foregoing, a *prima facie* case of anticipation has not been established and Ng does not anticipate Claim 1. **Claims 2-14, 16-20, and 22** depend, directly or indirectly, from Claim 1 and are patentable over the references of record for at least the same reasons as Claim 1. Furthermore, each of Claims 2-23 includes at least one other limitation that makes it further patentable over the references of record. However, due to the fundamental difference between Claim 1 and Ng discussed above, discussion of these additional differences is unnecessary and is foregone at this time. Withdrawal of the rejection of Claims 1-14, 16-20, and 22 under 35 U.S.C. §102(e) is requested.

Independent **Claim 24** recites similar limitations as Claim 1, specifically, regarding associating a data item with two category objects thereby causing the data item to be associated with a structure that includes storage for values for one or more attributes of each of the two category classes from which the category objects are instantiated. As shown above, *Ng* is absent a teaching or suggestion of the data item (e.g., the “order” object) being associated with a structure that includes storage for values for one or more attributes of a category class. Therefore, *Ng* is also absent a teaching or suggestion of the data item (e.g., the “order” object) being associated with a structure that includes storage for values for one or more attributes of each of multiple category classes (i.e., the data item inherits attributes from each of the first and second category classes).

Furthermore, Fig. 11A, which the Office Action relies on for creation of a second category object, depicts states of the object-relational mapping tool when updating an object model. In other words, Fig. 11A is about updating a relational database based on updates to a corresponding schema. This discussion in *Ng* has no relation to the category mechanism embodiment recited in Claim 24, where attributes corresponding to two different category classes, which are each different from the main class from which a data item is instantiated, descend to the data item. Claim 24 is amended as follows, to clarify that the category classes do not descend from the main class, i.e., neither of the category classes are subclasses of the main class:

wherein the first category class and the second category class are external to the class lineage of the class of which the data item is an instance.

Based on the foregoing, *Ng* does not anticipate Claim 24. **Claims 25-37 and 39-43** depend, directly or indirectly, from Claim 24 and are patentable over the references of record for at least the same reasons as Claim 24. Furthermore, each of Claims 25-45 includes at least

one other limitation that makes it further patentable over the references of record. However, due to the fundamental difference between Claim 24 and *Ng* discussed above, discussion of these additional differences is unnecessary and is foregone at this time. Withdrawal of the rejection of Claims 24-37 and 39-43 under 35 U.S.C. §102(e) is requested.

Rejections under 35 U.S.C. §103(a)

Claims 15, 21, 38, and 44 were rejected under 35 U.S.C. §103(a) as allegedly unpatentable over *Ng* in view of *Bennett*. This rejection is traversed.

Claims 15 and 21 depend from Claim 1 and Claims 38 and 44 depend from Claim 24. *Ng* is deficient in its teachings in regards to Claims 15, 21, 38 and 44 as it is in regards to the claims from which these claims respectively depend. Furthermore, *Bennett* does not cure the deficiencies in the teachings of *Ng*. Therefore, Claims 15, 21, 38, and 44 are patentable over the cited references of record. Withdrawal of the rejection of Claims 15, 21, 38, and 44 under 35 U.S.C. §103(a) is requested.

Claims 23 and 45 were rejected under 35 U.S.C. §103(a) as allegedly unpatentable over *Ng* in view of *Curtis*. This rejection is traversed.

Claims 23 and 45 depend from Claim 1 and Claim 24, respectively. *Ng* is deficient in its teachings in regards to Claims 23 and 45 as it is in regards to the claims from which these claims respectively depend. Furthermore, *Curtis* does not cure the deficiencies in the teachings of *Ng*. Therefore, Claims 23 and 45 are patentable over the cited references of record. Withdrawal of the rejection of Claims 23 and 45 under 35 U.S.C. §103(a) is requested.

CONCLUSION

For the reasons set forth above, it is respectfully submitted that all of the pending claims (1-45) are in condition for allowance. Therefore, the issuance of a formal Notice of Allowance is believed next in order, and that action is most earnestly solicited.

The Examiner is respectfully requested to contact the undersigned by telephone if it is believed that such contact would further the examination of the present application.

Please charge any shortages or credit any overages to Deposit Account No. 50-1302.

Respectfully submitted,

HICKMAN PALERMO TRUONG & BECKER LLP

Date: 6/8/05

John D. Henkhaus
John D. Henkhaus
Reg. No. 42,656

2055 Gateway Place, Suite 550
San Jose, CA 95110-1089
(408) 414-1080
Facsimile: (408) 414-1076

CERTIFICATE OF MAILING

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to: Mail Stop Amendment, Commissioner for Patents, P. O. Box 1450, Alexandria, VA 22313-1450

on 6/8/05 by Darci Sakamoto
Darci Sakamoto